THE CLAIMS

For the convenience of the Examiner, all pending claims of the present Application are shown below whether or not an amendment has been made.

1.   (Currently Amended) A method of identifying problems in applications, comprising:

monitoring at a kernel level system resource usage of one or more running applications without modifying run-time environments of the one or more applications;

from the system resource usage, determining a ~~system resource~~ **memory** usage pattern ~~of a first application~~ **for each of a plurality of applications,** ~~the system resource~~ **each memory** usage pattern indicating **for a selected one of the plurality of applications:**

a first memory usage amount during a first time period;

a second memory usage amount during a second time period; and

a change in the ~~system usage of the first application~~ **memory usage of the selected one of the plurality of applications** from ~~a first~~ **the first** time period to ~~a second~~ **a second** time period**, the change in the memory usage equaling the difference between the first memory usage amount and the second memory usage amount;**

determining whether ~~the change~~ **a change** in the ~~system resource~~ **memory** usage of ~~the first application~~ **a first application within the plurality of applications** satisfies a predetermined criteria associated with one or more problems; and

if the change in the ~~system resource~~ **memory** usage of the first application satisfies the predetermined criteria, identifying the first application to a user.

2.   (Previously Presented) The method of claim 1, wherein the system resource usage comprises a number of processes that each of the one or more applications have spawned and the predetermined criteria comprises exceeding a predetermined limit on the number of processes that each of the one or more applications may spawn.

3.   (Currently Amended) The method of claim 1, wherein:

monitoring at a kernel level system resource usage of one or more running applications comprises monitoring a parent-child relationship between one or more running processes and each of the one or more applications; and

determining whether the system usage pattern of the first application satisfies a predetermined criteria associated with one or more problems comprises determining whether the first application has orphaned one of the one or more running processes, **wherein an orphaned process is one wherein a child process is running even though a corresponding parent process associated with the child process is not running**.

4.   (Previously Presented) The method of claim 1, wherein the system resource usage comprises memory usage of the one or more running applications.

5.   (Currently Amended) The method of claim 1, wherein:

the one or more applications comprise one or more user applications initiated at the user level; and

monitoring at a kernel level system resource usage of one or more running applications comprises monitoring at a kernel level system resource usage of one or more running processes belonging to one or more user applications wherein the one or more running processes comprise one or more processes initiated at the kernel level by the one or more user applications, **the system resource usage comprising kernel space memory used by each of the plurality of user applications**.

6.   (Previously Presented) The method of claim 1 wherein the system resource usage of the one or more running applications is monitored over a plurality of consecutive discrete time periods.

7.   (Currently Amended)   ~~The method of claim 6, wherein:~~ A method of identifying problems in applications, comprising:

monitoring at a kernel level system resource usage of one or more running applications without modifying run-time environments of the one or more applications, wherein the system resource usage of the one or more running applications is monitored over a plurality of consecutive discrete time periods;

determining a system resource usage pattern of a first application, the system resource usage pattern indicating a change in the system usage of the first application from a first time period to a second time period, the system resource usage comprises an amount of memory usage for each of the one or more applications;

determining whether the change in the system resource usage of the first application satisfies a predetermined criteria associated with one or more problems, the predetermined criteria is a limit on a number of memory increases allowed during the plurality of time periods; and

if the change in the system resource usage of the first application satisfies the predetermined criteria, identifying the first application to a user.

8.   (Previously Presented) The method of claim 6, wherein:

the system resource usage comprises an amount of memory usage for each of the one or more applications; and

the predetermined criteria is a generally continuous increase in the amount of memory usage during the plurality of time periods.

9.   (Currently Amended) The method of claim 6, wherein:

the system resource usage comprises a number of processes that each of the ~~one or more~~ plurality of applications have spawned; and

the predetermined criteria is a generally continuous increase in the number of child processes spawned during the plurality of time periods.

10.　　(Currently Amended)　The method of claim 1, wherein identifying the first application to a user comprises saving an identifier of the first application in a reference file, and further comprising saving identifiers of any other of the ~~one or more~~ **plurality of** applications whose system usage pattern satisfies a predetermined criteria associated with one or more problems in the reference file.

11.　　(Currently Amended)　The method of claim 10, wherein a computer automatically:

monitors the kernel level system resource usage of one or more running applications;

determines whether a ~~system~~ **memory** usage pattern of a first application satisfies a predetermined criteria associated with one or more problems; and

identifies the first application.

12.　　(Currently Amended)　A method of identifying memory problems in applications, comprising:

monitoring at a kernel level memory usage of ~~one or more~~ **a plurality of** running applications without modifying run-time environment of the applications**, the memory usage indicating for each of the plurality of applications:**

　　　　a first memory usage amount during a first time period;

　　　　a second memory usage amount during a second time period; and

producing an output comprising at least the memory usage; and

determining a change in the memory usage of a first application **within the plurality of applications,** ~~the memory usage pattern~~ **the change in the memory usage** indicating a change in the memory usage of the first application from a first time period to a second time period**, the change in the memory usage equaling the difference between the first memory usage amount and the second memory usage amount**;

determining whether the change in the memory usage of the first application satisfies a predetermined criteria associated with one or more problems; and

if the change in the memory usage of the first application satisfies the predetermined criteria, identifying the first application to a user.

13.   (Previously Presented) The method of claim 12, wherein:

the memory usage of the one or more running processes is monitored over a plurality of consecutive discrete time periods, and

the predetermined criteria is a limit on a number of memory increases allowed during the plurality of time periods.

14.   (Previously Presented) A system for identifying memory problems in applications, comprising a computer operable to:

monitor at a kernel level memory usage of one or more running applications without modifying run-time environments of the one or more applications;

produce an output comprising at least the memory usage of one or more applications;

determining a memory usage pattern of a first application, the memory usage pattern indicating a change in the memory usage of the first application from a first time period to a second time period;

determine whether the change in the memory usage of the first application satisfies a predetermined criteria associated with one or more problems; and

if the change in the memory usage of the first application satisfies the predetermined criteria, identify the first application by saving an identifier of the first application in a reference file.

15. (Currently Amended) A method of identifying memory problems in applications, comprising:

monitoring at a kernel level memory usage of ~~one or more~~ **a plurality of** running applications without modifying run-time environments of the running applications**, the memory usage indicating for each of the plurality of applications:**

a first memory usage amount during a first time period; and

a second memory usage amount during a second time period;

determining a memory usage pattern of a first application **within the plurality of applications**, ~~the memory usage pattern~~ **the change in the memory usage** indicating a change in the memory usage of the first application from a first time period to a second time period**, the change in the memory usage equaling the difference between the first memory usage amount and the second memory usage amount**;

determining whether the change in the memory usage of the first application satisfies a predetermined criteria associated with one or more problems; and

if the change in the memory usage of the first application satisfies the predetermined criteria, identifying the first running application without identifying the one or more running applications whose memory usage patterns do not satisfy the predetermined criteria associated with the one or more memory problems.

16. (Original) The method of claim 15, wherein the monitored memory usage comprises at least a stack memory, data memory, and text memory.

17.   (Currently Amended)   A method of identifying memory problems in applications, comprising:

collecting system resource usage at a kernel level of one or more running processes belonging to one or more applications without modifying run-time environments of the one or more user running applications;

from the system resource usage, determining a ~~system resource~~ **memory** usage pattern of ~~a first application~~ **each of a plurality of applications,** ~~the system resource~~ **each memory** usage pattern indicating **for a selected one of the plurality of applications:**

a first memory usage amount during a first time period;

a second memory usage amount during a second time period; and

a change in the ~~system resource usage of the first application~~ **memory usage of the selected one of the plurality of applications** from ~~a first~~ **the first** time period to ~~a second~~ **the second** time period, **the change in the memory usage equaling the difference between the first memory usage amount and the second memory usage amount;**

determining whether ~~the change in the system resource usage of the first application~~ **a change in the memory usage of a first application within the plurality of applications** satisfies a predetermined criteria associated with one or more problems; and

if the change in the ~~system resource~~ **memory** usage of the first application satisfies the predetermined criteria, identifying the first application to a user.

18.   (Currently Amended)   A system for identifying problems in applications, comprising:

a data collection module operable to retrieve information about a plurality of running user-applications at a kernel level**, the information comprising:**

a first memory usage amount during a first time period;

a second memory usage amount during a second time period; and

a change in the memory usage of the selected one of the plurality of running user applications from the first time period to the second time period, the change in the memory usage equaling the difference between the first memory usage amount and the second memory usage amount; and

a data analysis module operable to:

determine abnormal system usage pattern in the information based on ~~changes in a system resource usage of the user-applications~~ **the difference between the first memory usage amount and the second memory usage amount**; and

identify a first user application **associated with the abnormal usage pattern** ~~whose system usage pattern corresponds to a change in a system resource usage~~ that satisfies a predetermined criteria associated with one or more problems.

19.   (Currently Amended)   A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform method steps of identifying problems in applications, comprising:

monitoring at a kernel level system resource usage of one or more running applications without modifying run-time environments of the one or more applications;

from the system resource usage, determining a ~~system resource~~ **memory** usage pattern ~~of a first application~~ **for each of a plurality of applications,** ~~the system resource~~ **each memory** usage pattern indicating **for a selected one of the plurality of applications:**

a first memory usage amount during a first time period;

a second memory usage amount during a second time period; and

a change in the ~~system usage of the first application~~ **memory usage of the selected one of the plurality of applications** from ~~a first~~ **the first** time period to ~~a second~~ **a second** time period**, the change in the memory usage equaling the difference between the first memory usage amount and the second memory usage amount**;

determining whether ~~the change~~ **a change** in the ~~system resource~~ **memory** usage of ~~the first application~~ **a first application within the plurality of applications** satisfies a predetermined criteria associated with one or more problems; and

if the change in the ~~system resource~~ **memory** usage of the first application satisfies the predetermined criteria, identifying the first application to a user.

20.   (Currently Amended)   The program storage device of claim 19, wherein the system resource usage comprises a parent-child relationship between one or more processes and each of the one or more applications; and

determining whether a system usage pattern of a first application satisfies a predetermined criteria associated with one or more problems comprises determining whether the first application has orphaned one of the one or more running processes**, wherein an orphaned process is one wherein a child process is running even though a corresponding parent process associated with the child process is not running**.

21. (Currently Amended) The method of Claim 1, wherein determining whether a ~~system~~ **memory** usage pattern of ~~a first~~ **the first** application satisfies a predetermined criteria associated with one or more problems comprises:

comparing the ~~monitored system~~ **memory** usage for the first application against the predetermined criteria; and

selecting the first application from the one or more running applications if the ~~system~~ **memory** usage pattern of the first application satisfies the predetermined criteria.

22. (New) The method of Claim 1, wherein each memory usage pattern for each of the plurality of applications comprises:

the amount of memory in a first region being used by a process to store text;

the amount of memory in a second region being used by the process to store data; and

the amount of memory in a third region being used by the process to store stack memory.

23. (New) The method of Claim 1, wherein:

monitoring system resource usage comprises monitoring memory usage by each of the plurality of applications over a plurality of consecutive discrete time periods;

determining whether a change in the memory usage of the first application comprises determining that an amount of memory usage for the first application has increased during a plurality of consecutive discrete time periods; and

diagnosing a memory leak associated with the first application in response to determining that the amount of memory usage for the first application has increased during the plurality of consecutive discrete time periods.

24. (New) The method of Claim 7, further comprising:

determining that the amount of memory usage for the first application has increased during a plurality of consecutive discrete time periods; and

diagnosing a memory leak associated with the first application in response to determining that the amount of memory usage for the first application has increased during the plurality of consecutive discrete time periods.

25.   (New) The method of Claim 12, wherein the memory usage of the one or more running applications comprises:

the amount of memory in a first region being used by a process to store text;

the amount of memory in a second region being used by the process to store data; and

the amount of memory in a third region being used by the process to store stack memory.

26.   (New) The method of Claim 12, wherein:

monitoring memory usage comprises monitoring memory usage by each of the plurality of running applications over a plurality of consecutive discrete time periods;

determining whether a change in the memory usage of the first application comprises determining that an amount of memory usage for the first application has increased during a plurality of consecutive discrete time periods; and

diagnosing a memory leak associated with the first application in response to determining that the amount of memory usage for the first application has increased during the plurality of consecutive discrete time periods.

27.   (New) The system of Claim 14, wherein the memory usage of the one or more running applications comprises:

the amount of memory in a first region being used by a process to store text;

the amount of memory in a second region being used by the process to store data; and

the amount of memory in a third region being used by the process to store stack memory.

28. (New) The system of Claim 14, wherein:

monitoring memory usage comprises monitoring memory usage by each of the plurality of running applications over a plurality of consecutive discrete time periods;

determining whether a change in the memory usage of the first application comprises determining that an amount of memory usage for the first application has increased during a plurality of consecutive discrete time periods; and

diagnosing a memory leak associated with the first application in response to determining that the amount of memory usage for the first application has increased during the plurality of consecutive discrete time periods.

29. (New) The program storage device of Claim 19, wherein the memory usage of the one or more running applications comprises:

the amount of memory in a first region being used by a process to store text;

the amount of memory in a second region being used by the process to store data; and

the amount of memory in a third region being used by the process to store stack memory.

30. (New) The program storage device of Claim 19, wherein:

monitoring memory usage comprises monitoring memory usage by each of the plurality of running applications over a plurality of consecutive discrete time periods;

determining whether a change in the memory usage of the first application comprises determining that an amount of memory usage for the first application has increased during a plurality of consecutive discrete time periods; and

diagnosing a memory leak associated with the first application in response to determining that the amount of memory usage for the first application has increased during the plurality of consecutive discrete time periods.